KNU OS Lab. 운영체제 연구실

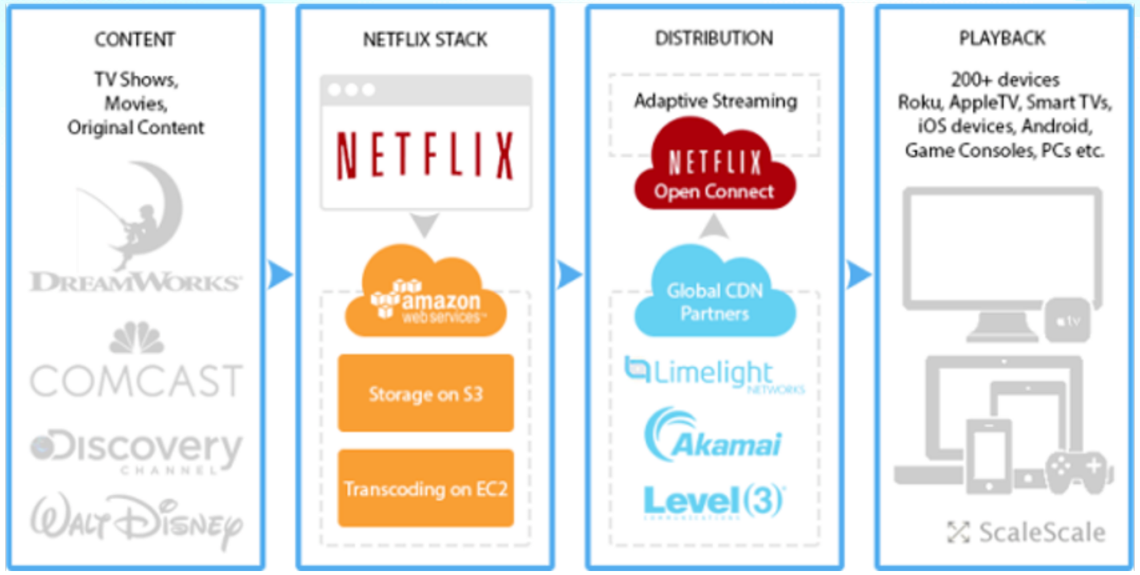
MicroVM on Edge: Is It Ready for Prime Time?

Kyungwoon Lee
Kyungpook National University

2025. 5. 30

Virtualization Technology

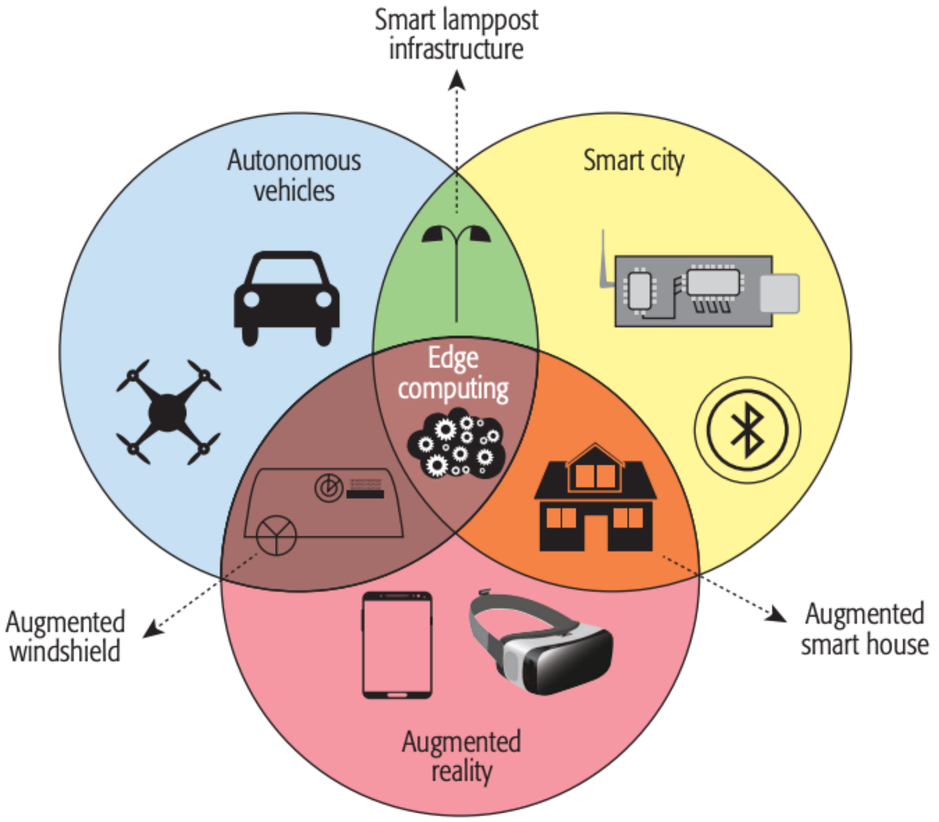
- Key enabling technology in cloud data centers
  - e.g., virtual machines (VM) and containers
  - Use cases: Netflix on Amazon AWS
    - On-premise → cloud data center (AWS)
      - Since 2016
    - VM performs video encoding



- Spread over diverse industry fields
  - Mobile devices, industrial IoT, automotive

Virtualization and Edge Computing

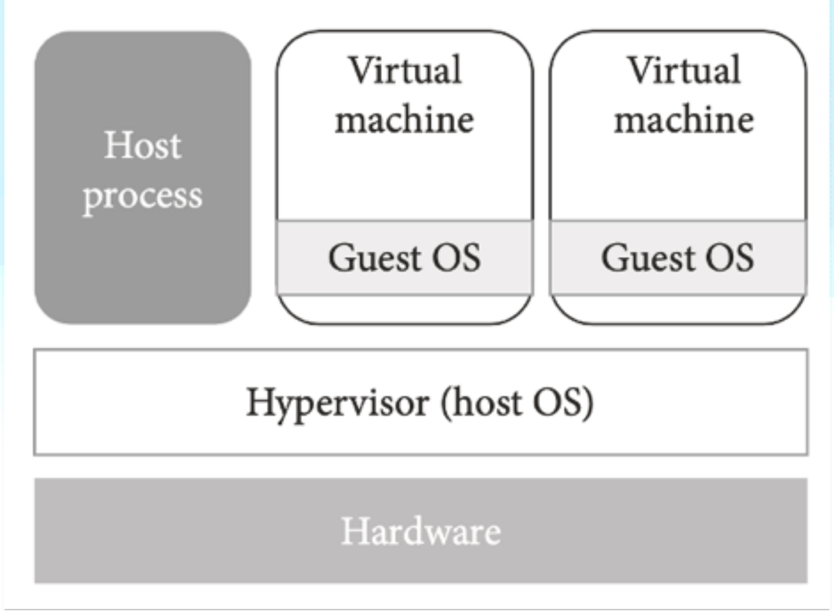
- Edge computing exploits virtualization techniques to offer various services
  - For satisfying service requirements such as security, scalability, and multi-tenancy*
  - VM, containers, and microVMs
- VMware already has built hyper-converged hardware to deploy at the edge**



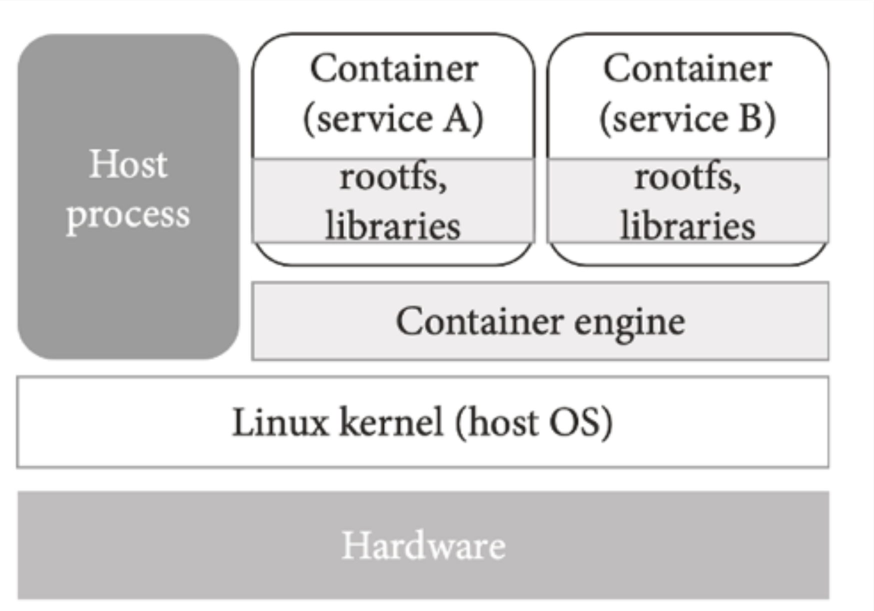
*"Consolidate IoT Edge Computing with Lightweight Virtualization", Roberto Morabito et. al., IEEE Network, 2018
**https://www.techtarget.com/searchitoperations/feature/Edge-virtualization-tools-make-strides-in-the-edge-market

Virtual Machines and Containers

- Virtual machines (VM)
  - Independent OS (Guest OS) and hardware abstraction
  - Need to access H/W through host OS
  - e.g., Xen, KVM, VMware ESXi, Windows Hyper-V

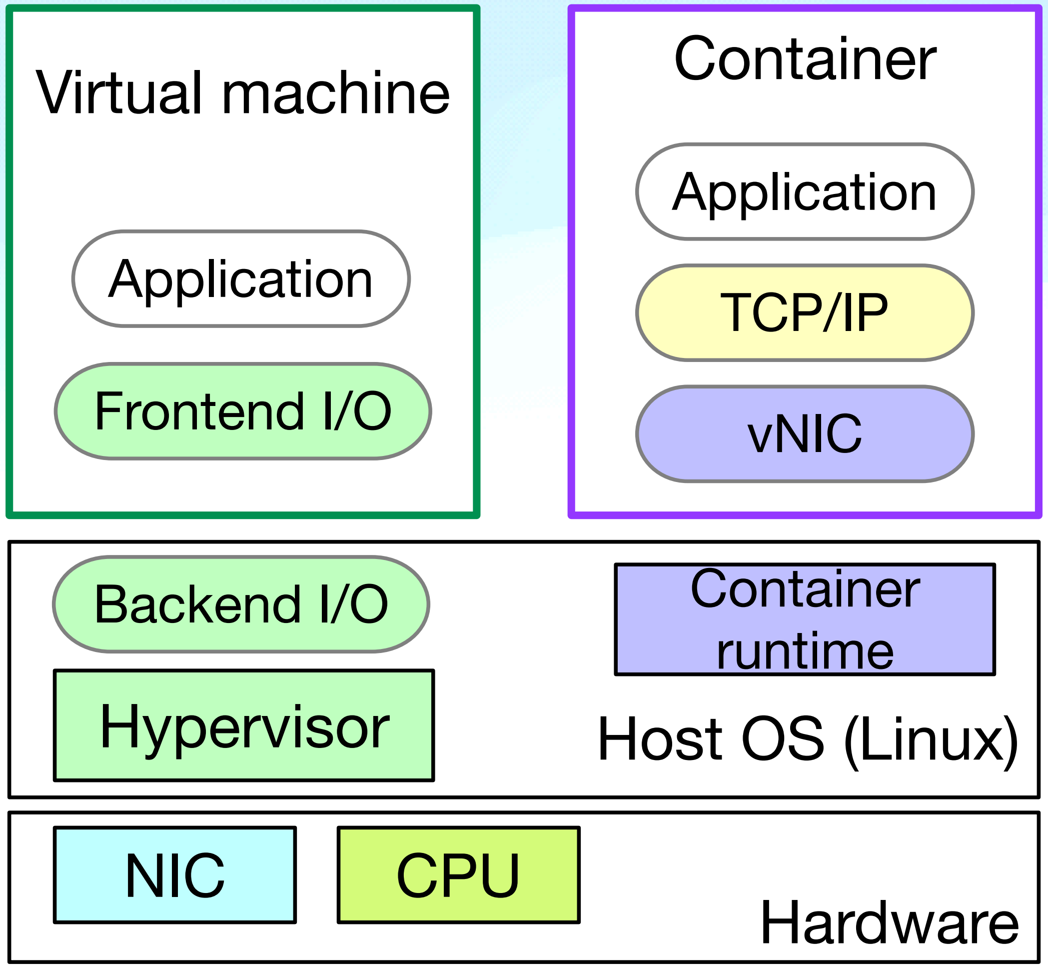


- Containers
  - Run as user-level processes
  - Share a single host OS
  - Independent root filesystem and library



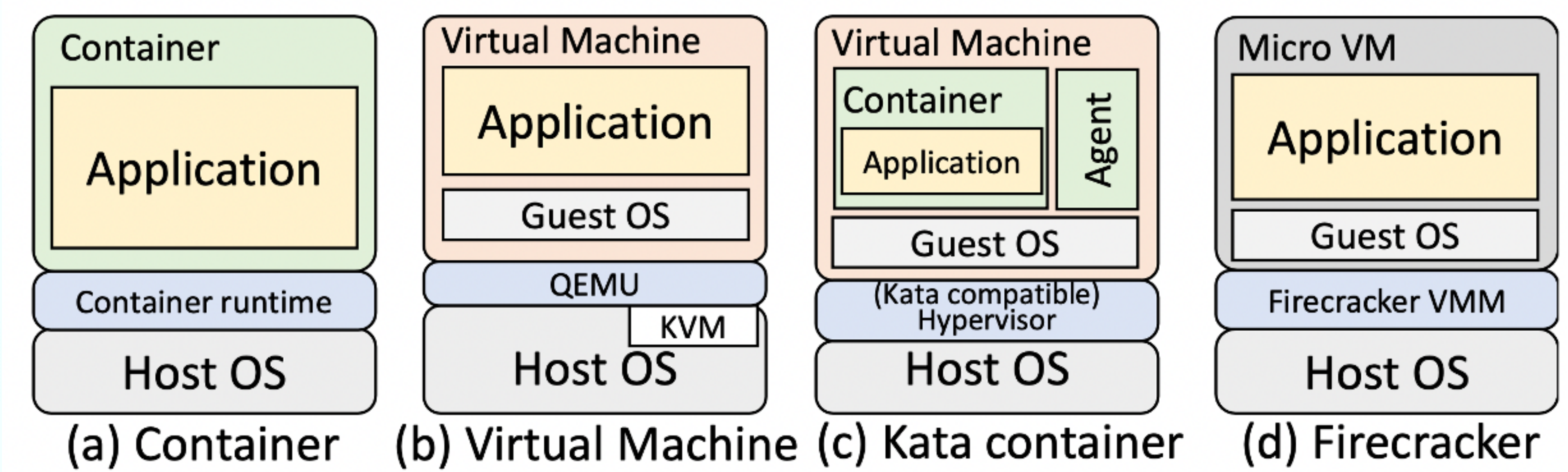
Research Area

- Application performance management for VMs and containers
- Network bandwidth or CPU scheduling
- Implementation based on Linux kernel



MicroVM Technologies

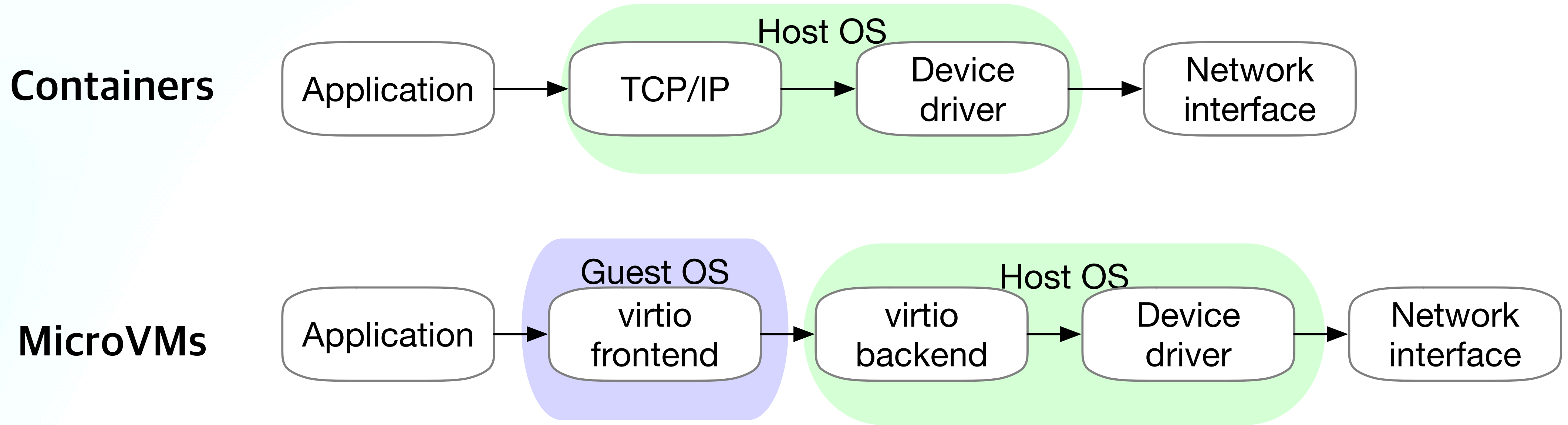
- Address the security concerns of containers
  - Focusing on providing stronger isolation capability
- Each microVM has an independent guest OS kernel
  - More lightweight than conventional VMs by various optimizations



MicroVM I/O Mechanism

- I/O handling using virtio driver model
  - Different from the direct execution of containers

- Various configurations pose impact on I/O performance
  - E.g., memory management policies, I/O buffering, or I/O scheduling policies



Motivation

- Most of studies on microVMs focus on cloud datacenter
    - No case for the edge environment
- Necessary to learn the adequacy of microVMs for handling edge workloads on edge devices
    - How do microVMs compare against containers in terms of I/O performance in general?
    - What are the performance characteristics of microVMs that differ from the containers in a fundamental way?

Research Objectives

- Thorough benchmark measurement on the **file and network I/O**
  - Compare native, container, and two microVM techniques on four state-of-the-art edge devices
- Investigation on the **relationship between the CPU usage and power consumption** of containers and microVMs
  - Quantify the overhead and the efficiency
- **In-depth system-level investigation**
  - Broaden the understanding of microVMs against containers

Experimental Setup

Devices

Name		Nano	Pi
Device model		Jetson Nano	Raspberry Pi 4 Model B
H/W	CPU	ARM Cortex-A57@1.43 GHz	ARM Cortex-A72@1.5 GHz
	Memory	2GB & 4GB	4GB & 8GB
	Storage	16 GB eMMC 5.1 204 MHz	micro-SD 50MHz
	Network	Realtek Gigabit Ethernet	BLE Gigabit Ethernet
S/W	Ubuntu/Linux	18.04 / 4.9	22.04 / 4.19 or 5.15
I/O device performance	Read	86.3 Mb/s	45.8 Mb/s
	Write	17.0, 21.6 Mb/s	11.7 , 17.4 Mb/s

Experimental Setup

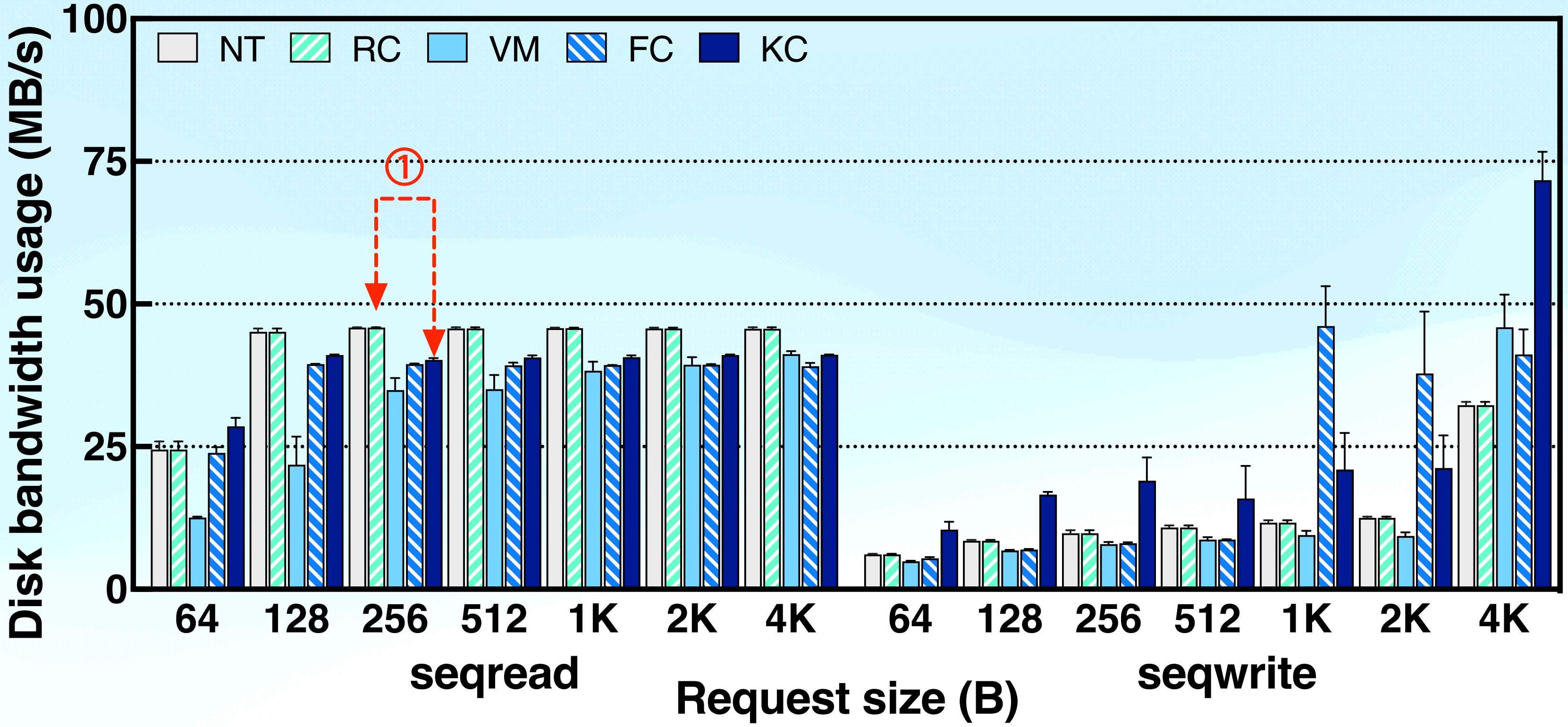
MicroVMs

Code	Details	File I/O	Network I/O
NT	Direct execution without any virtualization applied	ext4	Device-specific
RC	Docker container spawned by **runc** runtime	OverlayFS	Bridged networking
VM	Conventional **VM** using KVM/QEMU	virtio-blk	vhost-net
FC	MicroVM created on top of the **Fire-cracker** microhypervisor	virtio-blk (in Rust)	Virtio-based networking (in Rust)
KC	**Kata container** microVM	devicemapper	vhost-net

Benchmarks Used

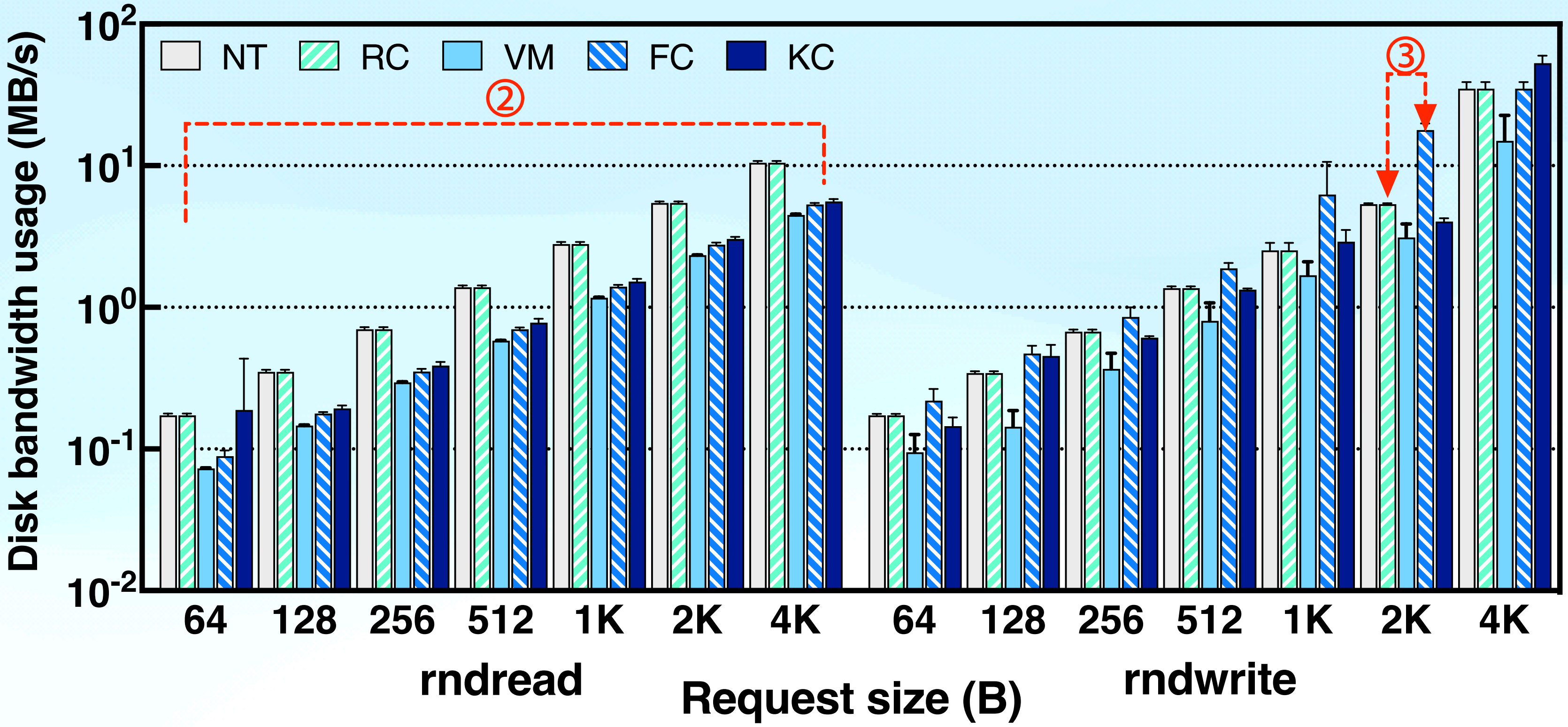
Experiments	Category	Name	Metrics	Details
File I/O performance	Micro-benchmark	**fio**	Throughput (MB/s)	Sequential/random read/ write operations
	Macro-benchmark	**Filebench**	Throughput (MB/s)	Mixed file operations (i.e., create, delete, append, read, write)
Network I/O performance	Micro-benchmark	**netperf**	Transmission rate (Mb/s)	TCP traffic for different message sizes (64B~4096B)
Power consumption	**Electricity usage monitor**		Actual power consumption (Watt)	
CPU utilization	System utility	**mpstat**	Average CPU usage (%)	User, System, Softirq, and Guest categories
		pidstat		Each process

Sequential Read/Write on Pi4



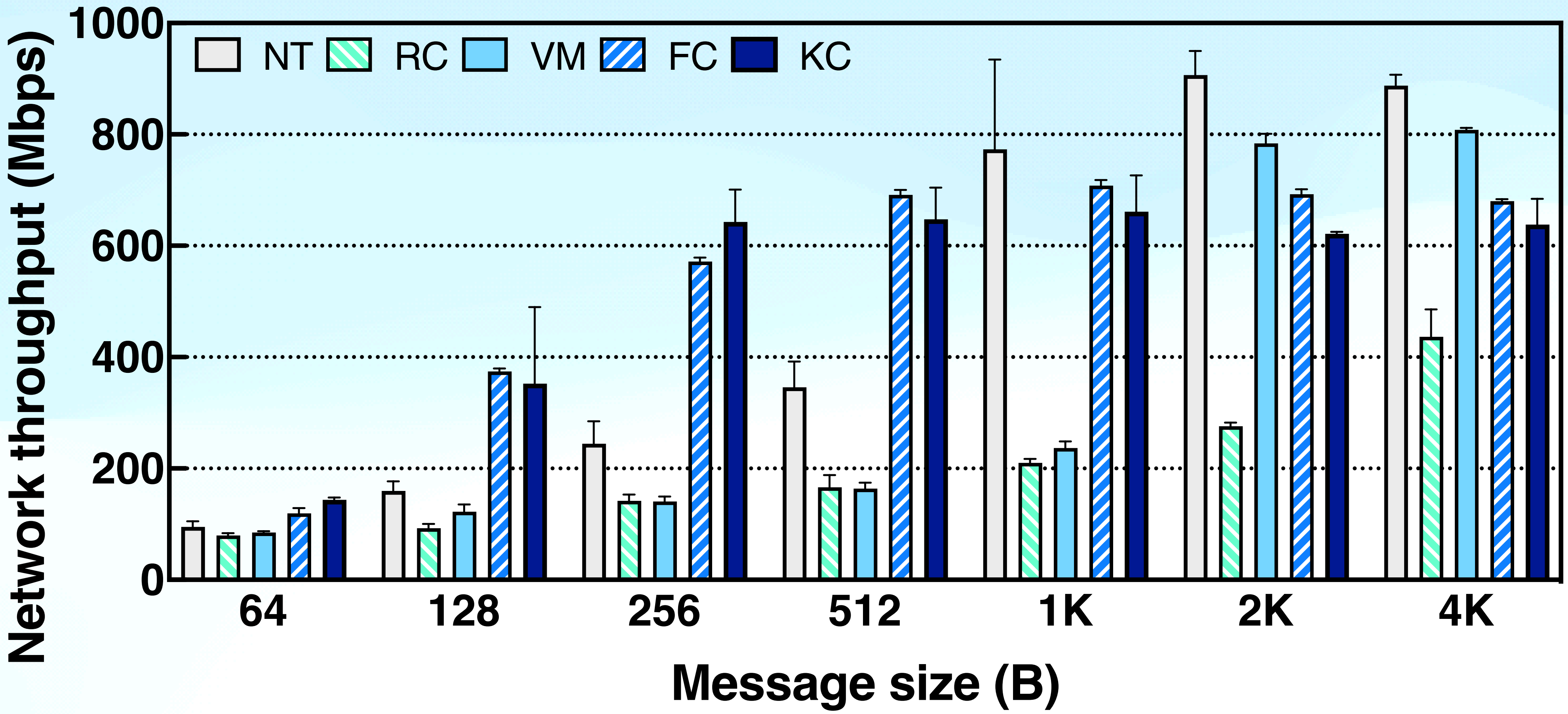
- Read: Performance gap is insignificant
  - Worst case(①): RC > KC (12%) for 256B requests
- Write: MicroVMs > container
  - Because the write operations are not directly applied to the storage device but handled in memory

Random Read/Write on Pi4



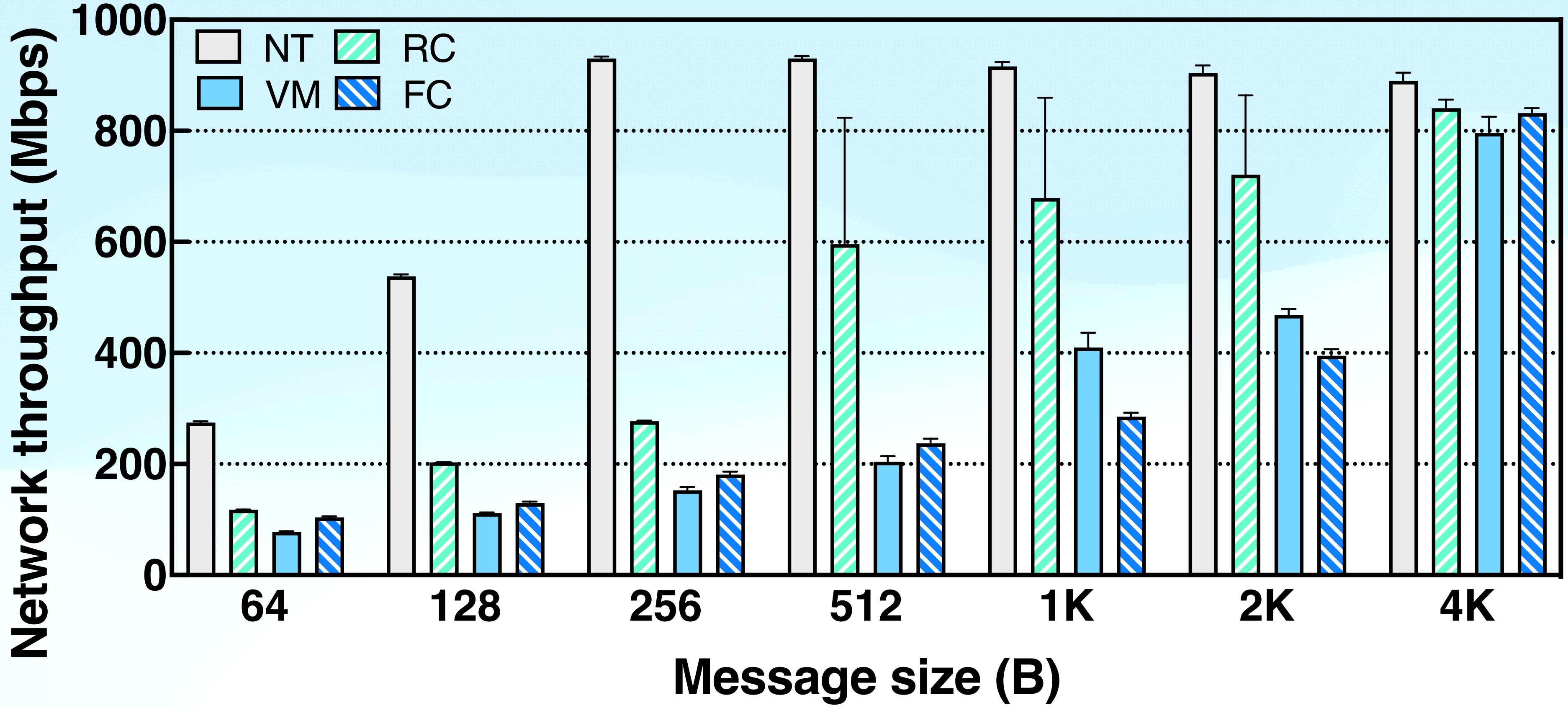
- Read(②): MicroVMs << container
  - Because of the small batch size of the fio benchmark (seqread: 512KB, rndread: 4KB)
- Write: RC < FC (3.3x) for 2KB requests
  - Similar to sequential write operations, the virtual block devices of microVMs do not persist the data to the storage devices

Network Performance on Pi4



- MicroVMs can deliver a comparable or better network performance than containers
  - Thanks to the virtual network device, virtio_net
    - Handle consecutive packets as a group → Large transmission size

Network Performance on Nano2



- The performance of microVMs decreases compared to Pi4
  - Because of the different network devices and offloading techniques (e.g., TSO) → Boost up the throughput of NT and RC
- MicroVMs require high CPU usage for system-level processing
  - Limited network throughput

Implications

File Systems

- The performance gap between the containers and microVMs varies depending on the type of file operations
  - Sequential and random
    - The performance of microVMs on sequential workloads is comparable to other techniques
  - Read and write
    - MicroVMs can outperform the native and the containers significantly in random write workloads, but lag behind in random read workloads.

- Root cause is the virtual block device that does not directly apply the result of file operations

Implications

Network and energy consumption

- MicroVMs can deliver a comparable or better network performance than containers
  - The main reason is the batch processing of virtual network devices in microVMs
- MicroVMs consume more power than containers
  - This is because the power consumption is highly related to the CPU usage
  - However, in terms of efficiency, microVMs outperform containers by achieving high performance with low overhead

Suitability of MicroVMs

- We focus on the performance aspect
  - Performance level of microVMs is comparable to containers

- In terms of the security
  - It is known that microVMs have stronger isolation than containers
  - A study reported that microVMs can offer 4.2 to 7.5 times more security than containers

- Usability: some limitations
  - Lack of support for diverse hardware devices such as USBs and GPUs

Summary

- We assessed the suitability of microVMs for edge computing
- Our experimental results show that
  - MicroVMs do not exhibit significant performance degradation, but provide better performance for some cases
  - Additional power consumption is at 19%, much lower than the performance improvement (3x for network performance)
- There exists high value in adopting microVMs in edge computing
  - Superior isolation capability with the comparable performance to containers